AF
TFW

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In Re Application of:		)
Inventors:	Guy Eden	)
Serial No.:	09/859,660	) ATTORNEY FILE NO. SLA1014
Filed:	May 16, 2001	)
		) Customer No.: 55,286
Title:	SYSTEM AND METHOD FOR DISCOVERING NETWORK COMPONENTS	) Examiner: Ramsey Refai
		) Confirmation No.: 3934
		) Art Unit: 2152

CERTIFICATION UNDER 37 CFR § 1.8

I hereby certify that the documents referred to as enclosed herein are being deposited with the United States Postal Service as first class mail on this date 10/5/2006 in an envelope addressed to: Mail Stop Amendments, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.

Date 10/4/2006 Signature

Board of Patent Appeals and Interferences
United States Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450

RESPONSE TO NON-COMPLIANT APPEAL BRIEF

In response to a Notice of Non-Complaint Appeal Brief mailed September 28, 2006, a corrected Appeal Brief is enclosed.

The revised Appeal Brief provides additional citations to the Applicant's specification and drawings with respect to independent claim 15. The revised Appeal Brief has every thing from the Evidence Appendix, except the patent application cited by the examiner as prior art.

A Related Proceeding Appendix has been added, with no related proceedings.

Respectfully submitted,

Date: 10/4/2006

Gerald Maliszewski
Registration No. 38,054

Customer Number 55,286
P.O. Box 270829
San Diego, CA 92198-2829
Telephone: (858) 451-9950
Facsimile: (858) 451-9869
gerry@ipatentit.net

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In Re Application of:		
Inventors:	Guy Eden	
Serial No.:	09/859,660	ATTORNEY FILE NO. SLA1014
Filed:	May 16, 2001	
		Customer No.: 55,286
Title:	SYSTEM AND METHOD FOR DISCOVERING NETWORK COMPONENTS	Examiner: Ramsey Refai
		Confirmation No.: 3934
		Art Unit: 2152

Board of Patent Appeals and Interferences
United States Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450

BRIEF ON APPEAL

This is an appeal from the rejection by Examiner Ramsey Refai, Group Art Unit 2154, of claims 1-6, 8-20, and 22-26 as set forth in the CLAIMS APPENDIX, all claims in the application.

REAL PARTY IN INTEREST

The real party in interest is Sharp Laboratories of America, Inc., as assignee of the present application by an Assignment in the United States Patent Office on August 11, 2001, with a recordation date of May 16, 2001 at Reel 011840, Frame 0401.

RELATED APPEALS AND INTERFERENCES

None.

STATUS OF THE CLAIMS

Claims 7 and 21 are canceled. Claims 1-6, 8-20, and 22-26 are in the application.

Claims 1-6, 8-20, and 22-26 are rejected.

Claims 1-6, 8-20, and 22-26 are appealed.

STATUS OF AMENDMENTS

Section 1 of the Final Office Action objected to claims 15 and 24. The Applicant made a good faith effort to amend the claims in a response under 37 CFR 1.116, received at the PTO on April 18, 2006. Line 9 of claim 15 recites the phrase "to5", which the Applicant attempted to amend to the phrase "to". Line 3 of claim 24 recites the phase "a the", which the Applicant attempted to amend to the phase "the". An Advisory Action mailed on June 19 stated that the Applicant's response was found to be noncompliant. Therefore, these amendments have not been entered.

SUMMARY OF CLAIMED SUBJECT MATTER

The problem addressed by the present invention is presented in the specification at page 1, line 11, through page 3, line 12. Generally, the problem is associated with network discovery. Upon initialization, a querying device (i.e., a personal computer) conventionally checks its list of network-connected components (e.g., a printer) by attempting to communicate with every device on the list. Once the list has been checked, the querying device builds a graphical user interface (GUI) to show a user the connected devices actually available (see Fig. 1). The problem occurs when a device is no longer connected to the network, or is turned off. Then, the querying device can wait for as long as 30 seconds for a response from a single device. If no response is received, and a timeout occurs, the GUI indicates that the device is not connected (see Fig. 2). However, a conventional GUI does not create a display, which indicates device availability, until all the device queries have been resolved.

The Applicant's solution to the problem is simple. Rather than waiting for all the devices to reply, the querying device first builds a GUI representation of network-connected devices, see the timing diagram of Fig. 8. Then, as devices either respond or timeouts occur, the GUI representation (availability) of a device is modified. A process for querying network-connected devices to determine availability (claim 1) is described at page 15, ln. 9, through page 16, ln. 12. In its broadest form, Step 1304 builds a GUI representation of network-connected device availability. Then, Step 1306 begins querying devices. As described in dependent claims, Step 1305 shows that the devices may initially be represented in the GUI as unavailable. If a reply is received (Step 1308),

the GUI is revised to show the device as available. If a timeout occurs (Step 1312), the device unavailable status is maintained (Step 1314). The device status initially represented in the GUI is arbitrary, since the GUI is updated with actual values once the query process is completed.

A method for building a GUI that represents device availability, independent of system timeouts (claim 13), is described at page 17, ln. 24, through page 18, ln. 9 (see Fig, 14). Step 1402 builds a GUI representing network-connected devices. Step 1404 initially represents devices as unavailable. Step 1406 modifies the GUI representation to show a device as available in response to receiving a communication from that device.

The invention is recited from a systems/device perspective in claim 15, which is described at page 7, ln. 7, through page 8, ln. 2 (see Fig. 3). A querying device 102 has a GUI 104 representing network-connected devices and a network port on line 118. After building the GUI, the querying device sends a query to at least one network-connected device (e.g., device 106). The GUI representation of the network-connected devices is updated in response to sending the query.

GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

1. Whether claims 1-6, 8-20, and 22-26 are indefinite under 35 U.S.C. 112, second paragraph, for failing to particularly point out and claim the subject matter of the invention.

2. Whether claims 1-5, 12-19, and 25 are anticipated under 35 U.S.C. 102(e) by Carcerano et al. (US 6,308,205).

3. Whether claims 6, 8-11, 20, 22-24, and 26 are unpatentable under U.S.C. 103(a) over Carcerano et al. in view of admitted prior art (AAPA).

ARGUMENT

1. The rejection of claims 1-6, 8-20, and 22-26 under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and claim the subject matter of the invention.

Claim 13: The Final Office Action states that claim 13 has been rejected because of the recited term "network-connected device". The Office Action states that "(i)t is not clear if this device is a device from the group of "the network-connected devices" previously introduced in the preamble."

In addition to appearing in the preamble, the phrase "network-connected devices" appears in the first step of the claim, which recites "...building a GUI representation of network-connected devices..." Therefore, the issue to be resolved should be whether ambiguity exists between the step of "building the GUI representation of network-connected devices" and the step of "sending a query to a network-connected device".

Claim 13 can be summarized into three steps, which are:

building a GUI representation of network-connected devices;

sending a query to a network-connected device; and,

modifying the GUI representation of the network-connected device in response to sending the query.

Even if it is possible to imagine some ambiguity between "the network-connected device" in the second step, and "the network-connected devices" in the first step, this ambiguity would be resolved in the third step, in which the representation of "the network-connected device" is modified. Since the GIU representation of "the network-connected device" is modified, it must necessary be one of the GUI-represented "network-connected devices" in the first step.

Claim 15: The Final Office Action states that the term "at least one device" is indefinite because it is not clear to what this term is referring. Claim 15 is an independent claim reciting two basic network-connected elements: "a querying device", and "at least one device having a network connection port for communicating with the querying device". Thus, the claim clearly states that the "at least one device" is a device that is communicating with the querying device.

Claims 18-26: The Final Office Action states that it is not clear whether "a first network-connected device" and "a second-network-connected device" are from among "the network-connected devices" previously recited.

This rejection is clearly addressed by simply reading the claims. Claim 18 recites the steps of sending a query to "each of the network-connected devices" and "receiving a query reply from a first network-connected device". It is clear and unambiguous that the first network-connected device is a device that responds to the query sent to "each of the network-connected devices". Likewise, in claim 19, it is clear that the second network-connected device is a network-connected device that does not respond to a query.

Antecedent Basis: In claim 1, the Office Action states that there is no antecedent basis for the terms "network-connected devices", "the GUI", and "the queries". With respect to the term "network-connected device", the Applicant does not understand the rejection. The term is used many times throughout claim 1, and claims dependent from claim 1. The term is initially introduced in line 5 of claim 1.

With respect to the term "the GUI" (in line 7), the term "GUI representation" is initially introduced in lines 4 and 5. The Applicant respectfully submits that an expert reading claim 1 would find no ambiguity between the terms "building a GUI representation" in Step 1, and "following the building of the GUI" in Step 2.

With respect to the term "the queries" (in line 9), the phase "sending a query...to the network-connected devices" is initially introduced in line 6 of claim 1. The Applicant submits that an expert would find no ambiguity between "sending a query ...to...devices" and "the queries".

The Office Action states that there is no antecedent basis for the term "the GUI" in claims 2, 3, 12. As noted above, the term "building a GUI representation" is initially introduced in lines 4 and 5 of claim 1. Further, the phrase "following the building of the GUI" is presented on line 7 of claim 1.

The Office Action states that there is no antecedent basis for the term "the GUI representation" in claims 9 and 15. As noted above, the term "building a GUI representation" is initially introduced in lines 4 and 5 of claim 1.

The Office Action states that there is no antecedent basis for the terms "updating the GUI representation" in claims 4 and 5. The term

"updating the GUI representation" is initially introduced in line 9 of claim 1.

The Office Action states that there is no antecedent basis for the term "changing the GUI representation of the first network-connected device" in claims 4 and 9. The term "updating the GUI representation" is initially introduced in line 9 of claim 1. The first network-connected device is initially introduced in claim 4. Claim 4 further limits the step of "updating the GUI representation", initially presented in claim 1, as "changing the GUI representation of the first network-connected device". Alternately stated, this is the initial presentation of the substep of "changing the GUI representation of the first network-connected device".

The Office Action states that there is no antecedent basis for the term "the GUI representation of the second network-connected device" in claims 5 and 10. The term "updating the GUI representation" is initially introduced in line 9 of claim 1. The second network-connected device is initially introduced in claim 5. Claim 5 further limits the step of "updating the GUI representation", initially presented in claim 1, as "maintaining the GUI representation of the second network-connected device". Alternately stated, this is the initial presentation of the substep of "maintaining the GUI representation of the second network-connected device".

In claim 13, the Office Action states that there is no antecedent basis for the term "the network-connected devices" and "the GUI representation of the network-connected device". The term "network-connected devices" is initially introduced in line 6 of claim 13. The term "building a GUI representation" is initially introduced in lines 5 and 6.

In claim 14, the Office Action states that there is no antecedent basis for the term "modifying the GUI representation". In response, the Applicant notes that the term is initially presented in claim 13, line 9.

In claim 15, the Office Action states that there is no antecedent basis for the terms "network-connected devices" and "the GUI representation". The term "network-connected devices" is initially introduced in line 4 of claim 15, and the term "a GUI representing network-connected devices" in introduced in lines 3 and 4. The Applicant respectfully submits that an expert in the art would be able to identify "a GUI representing network-connected devices" as "the GUI representation of network-connected devices.

In claim 18, the Office Action states that there is no antecedent basis for the term "the GUI representation of the first network-connected device". The first network-connected device is initially introduced in claim 18. Claim 18 further limits the querying device's GUI representation of network-connected devices (first introduced in claim 15) by reciting that the GUI representation of the first device is changed to "available", as a result of the first device responding to a query.

In claim 20, the Office Action states that there is no antecedent basis for the term "the GUI representation of the second network-connected device". The second network-connected device is initially introduced in claim 19. Claim 20 (dependent from claim 19) further limits the querying device's GUI representation of network-connected devices (first introduced in claim 15) by reciting that the

querying device "maintains the GUI representation of the second device as unavailable", in response to the second device not responding to a query.

In claim 23, the Office Action states that there is no antecedent basis for the term "the querying device GUI" and "the representation of the first network-connected device". In response, the Applicant notes that claim 15 introduces the claim element of "a query device having a GUI representing network-connected devices". The Applicant submits that an expert in the art would not find any ambiguity between the term "the querying device GUI" and the querying device initially recited in claim 15. The term "GUI representation of the first network-connected device" is initially presented in claim 18, from which claim 23 depends.

In claim 24, the Office Action states that there is no antecedent basis for the term "a query reply", "the querying device GUI", and "the representation of the second network-connected device". In response, the Applicant does not understand how there can be an antecedent basis problem with the initial introduction of a phase prefaced with the article "a" (a query reply). Alternately stated, this is the initial introduction of a False answer received in response to a query reply.

The Applicant submits that an expert in the art would not find any ambiguity between the term "the querying device GUI" and the querying device initially recited in claim 15. Claim 19 recites "the GUI representation of the second network-connected device as unavailable". The Applicant submits that an expert would not find ambiguity between claim 19 and the term "the representation of the second network-connected device as unavailable" recited in claim 24.

The above-mentioned antecedent rejections appear to be spurious, pedantic, contradictory, and oblivious to the manner by which broad limitations in the base claim are further limited in the dependent claims. As support for this assertion, page 4 of the Office Action states that there is a difference between the limitations of "building a GUI representation of network-connected devices" and "building a GUI representing network-connected devices". The Office Action states that since claim 1 describes "building a GUI representation", any dependent claims that recite "building a GUI" are indefinite since a GUI has not been claimed. This analysis is incorrect on a number of levels.

First, 35 U.S.C. 112, second paragraph, states that the specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as the invention. The Applicant respectfully submits that a person skilled in the art would not find a contradiction between the terms "building a GUI representation" and "building a GUI".

Second, antecedent basis refers to the issue of ambiguity in the recitation of related claim elements. Neither 35 U.S.C. nor the MPEP equate definiteness with the rigid use of exactly the same word endings or exactly the same word order. "(T)he failure to provide explicit antecedent basis for terms does not always render a claim indefinite. If the scope of a claim would be reasonable ascertainable by those skilled in the art, then the claim is not indefinite. *Ex parte Porter*, 25 USPQ2d 1144, 1145 (Bd. Pat. App. & Inter. 1992).

We have held that a claim is not indefinite merely because it poses a difficult issue of claim construction; if the claim is subject to construction, i.e., it is not insolubly ambiguous, it is not invalid for

indefiniteness. *Honeywell Int'l, Inc. v. Int'l Trade Comm'n*, 341 F.3d 1332, 1338-39 (Fed. Cir. 2003). That is, if the meaning of the claim is discernible, "even though the task may be formidable and the conclusion may be one over which reasonable persons will disagree, we have held the claim sufficiently clear to avoid invalidity on indefiniteness grounds." *Exxon Research & Eng'g Co. v. United States*, 265 F.3d 1371, 1375 (Fed. Cir. 2001).

Third, several of the so-called antecedent problems refer to dependent claims where elements broadly claimed in the base claim are further limited in dependent claims. For example, a *first* network-connected device is introduced as an example of a network-connected device that responds to a query, and its GUI representation is further refined to recite that the GUI representation is changed from "unavailable" to "available" (e.g., claims 4 and 18).

2. The rejection of claims 1-5, 12-19, and 25 as anticipated under 35 U.S.C. 102(e) by Carcerano et al. (US 6,308,205).

Section 4 of the Office Action states that claims 1-5, 12-19, and 25 have been rejected under 35 U.S.C. 102(e) as anticipated by Carcerano et al. ("Carcerano"; US 6,308,205). With respect to claims 1 and 15, The Office Action states that Carcerano describes building a GUI representing available devices, and querying devices after building the GUI.

At col. 2, ln. 46-54, Carcerano describes a web browser that sends a request to network device. At col. 11, ln. 38-51, Carcerano

describes a browser-based network management system that sends URL-encoded requests to obtain and monitor the status of network devices. At col. 14, ln. 47-67, Carcerano describes Steps 811 and 812 of Fig. 8B. These steps describe receiving a HTTP response and receiving configuration data. The above-mentioned passages are cited in the Office Action as evidence that Carcerano describes the building of a GUI prior to sending queries to network-connected devices.

In the ***Response to Arguments*** Section, on page 9, the Office Action states that Carcerano builds a browser interface prior to sending device status inquiries. More specifically, the Office Action states that Carcerano teaches that the data to fill a template, which is used to construct the interface, is obtained from a database, citing col. 2, ln. 46-54, col. 11, ln. 38-51, and col. 14, ln. 47-67.

Generally, the Office Action appears to be merging the step of sending of queries from users, to a database of device configuration information, with the separate step of sending queries by the database to the network-connected devices, to collect the device configuration data.

Col. 2, ln. 35-54 states:

> Accordingly, in one aspect, the invention is a system that allows a remote network user to view and update a configuration of at least one of a plurality of network devices connected to a network, by using a web browser on the user's workstation. The system repeatedly polls each of the network devices over the network for configuration information. The configuration information is stored in a database. A first URL-encoded request is received from a user's workstation, preferably using a standard web browser communicating using HTTP. The first request identifies a targeted one of the network devices, together with a request for the targeted device's configuration. Responsive to the first request, a response corresponding to the requested

> configuration is generated dynamically from the database, with the response preferably being in a format representative of a visual display of configuration information for the targeted network device. The response is preferably dynamically-generated HTML code based at least in part on the configuration information stored in the database and on a template.

In summary, the above-quoted section from the Carcerano disclosure describes a system that repeatedly polls network devices for configuration data. This information is stored in a database on a template. This passage also describes supplying device configuration data from the database, as a visual display, to requesting users. However, the passage does not describe a GUI or visual display depicting all the devices in the network. The user merely receives a visual display for a particular requested device. More important, the passage does not describe a database, or GUI representation of a database, that is built prior to sending a query to a device. First, the passage clearly states that the template in the database is only filled after "repeatedly polling" a device. Second, user queries are not sent to, or answered by the device itself, they are answered by the database. Therefore, regardless of whether Carcerano is viewed from the perspective of a user making inquiries to the database, or the database making inquiries to the network-connected devices, Carcerano does not describe a GUI representation (or even a database) that is built before sending queries to the network devices.

Col. 11. ln. 38-51 states:

> Browser-based network management system 109 communicates with a requesting station such as workstation 70 using HTTP. In order to obtain and monitor status and configuration of a managed device (or the network interface device for that managed device), browser 83 on workstation 70 sends a URL-encoded request for status or

configuration information about a managed network device on network 1. In response, HTTP server 103 accesses the CGI script identified in the URL-encoded request so as to dynamically generate a response representative of a visual display of the status and configuration. This response is generated by filling in one of templates 107 with data from database 105. The response is communicated to browser 83, which displays the visual display.

This passage is similar to the section cited from col. 2. A user's browser 83 sends a request for status or configuration data about a device. A server 103 accesses a script from an already filled template. The response is displayed on browser 83. Once again, this section merely describes the accessing of a database. At this point in Carcerano's process, the database has already completed its device inquiries, which result in the database templates being filled.

Col. 14, ln. 47-67 states:

If such a URL-encoded request has been received, flow proceeds to step S811, where HTTP server 103 dynamically generates a response, preferably in the form of HTML code. This HTML code is generated by accessing the CGI script (or ASP web page) identified by the URL. Then, HTTP server 103 generates the HTML code based on the CGI script (or ASP web page) and the entries in database 105 for the device identified in the URL-encoded request. In the case of a CGI script, HTTP server 103 executes that script, which accesses one of templates 107 in dependence on the nature of the request and accesses database 105 so as to complete the template. Then, the HTML code for the completed template is returned to browser 83 through HTTP server 103.

In step S812, it is determined if HTTP server 103 has received a URL-encoded request from browser 83 with an update to configuration data. If such a URL-encoded request has been received, flow proceeds to step S813, where a CGI script identified by the request is executed so as to update database 105 with the updated configuration data. This update is placed in a queue in the database so that the process of FIG. 8A can identify the change.

Again, this cited passage is similar to the passages from Carcerano already quoted. As in the passage of col. 11, the above-described process describes occurrences that take place after the database makes device inquiries, and fills the templates as a result of these inquiries. The scripts are dynamically supplied to users as a visual display, by the database, when a user requests configuration data for a particular device.

Support for the Applicant's position can also be found in Carcerano's Fig. 8A. The first step of Fig. 8A (Step S801) describes "discover devices". Step S802 describes "poll network devices". Step S803 describes "store configuration information in database". At col. 13, ln. 58-65, Carcerano describes the use of conventional SNMP or DMI protocol to perform discovery (Step S801). Alternately stated, Carcerano begins his process with a conventional device discovery operation. A conventional discovery operation is described in the Background Section of the Applicant's specification (Evidence Appendix Attachment A, pages 1-3. The Applicant's claims are a solution to the time-out problem associated with conventional device discovery.

Claims 1, 13, and 15 of the claimed invention describe initially building a GUI representation of network-connected devices. Only after building the GUI does the querying device send a query to network-connected devices. Carcerano does not disclose a database that is built prior to making device inquiries. Therefore, Carcerano cannot disclose a visual representation of network-connected devices that is built prior to sending device status inquiries.

"A claim is anticipated only if each and every element as set forth in the claim is found, either expressly or inherently described, in a

single prior art reference." *Verdegaal Bros. v. Union Oil of California*, 814 F.2d 628, 631, 2 USPQ2d 1051, 1053 (Fed. Cir. 1987).

Carcerano does not explicitly describe every limitation of claims 1, 13, and 15. Since Carcerano does not describe every limitation of the claimed invention, he cannot anticipate. Claims 2-5 and 12, dependent from claim 1, claim 14, dependent from claim 13, and claims 16-19 and 25, dependent from claim 15, enjoy the same distinctions from the Carcerano reference.

3. The rejection of claims 6, 8-11, 20, 22-24, and 26 as unpatentable under U.S.C. 103(a) over Carcerano et al. in view of admitted prior art (AAPA).

In Section 15 of the Office Action claims 6, 8-11, 20, 22-24, and 26 have been rejected under 35 U.S.C. 103(a) as unpatentable with respect to Carcerano, in view AAPA. With respect to claims 9-10 and 23-24, the Office Action states that Carcerano fails to teach True/False answer, but that it would have been obvious to combine the True/False answers taught in the AAPA with Carcerano "to provide a method of querying devices for status information by labeling a device as unavailable if the device replies to a query and unavailable if the device fails to respond." With respect to claim 11, the Office Action states that Carcerano teaches spawning a thread to a network-connected device such as a printer, copier, scanner, or the like. With respect to claim 26, the Office Action states that Carcerano teaches a refresh command.

An invention is unpatentable if the differences between it and the prior art would have been obvious at the time of the invention. As

stated in MPEP § 2143, there are three requirements to establish a *prima facie* case of obviousness.

> First, there must be some suggestion or motivation, either in the references themselves or in the knowledge generally available to one of ordinary skill in the art, to modify the reference or to combine reference teachings. Second, there must be a reasonable expectation of success. Finally, the prior art reference (or references when combined) must teach or suggest all the claim limitations. The teaching or suggestion to make the claimed combination and reasonable expectation of success must both be found in the prior art and not based on applicant's disclosure. *In re Vaeck* 947 F.2d 488, 20 USPQ2d, 1438 (Fed. Cir. 1991).

Beginning at page 1, ln. 25, the Applicant's specification states that conventional systems "build the GUI to validate device availability only *after* (emphasis added) it has received replies from all the components (network devices) whose existence the application wants to query." This conventional process is described in more detail in the explanation of Fig. 1, where it states that Step 12 sends queries to network-connected devices, and Step 16 waits for all the queries (threads) to return with an answer. Only after these 2 steps are taken, is the GUI built in Step 18 (page 2, ln. 21. through page 3, ln. 4). It takes as long as 30 seconds for a time-out to occur, if a device does not respond to a query. Due to the time-out problem, status updates can be delayed as long as 30 seconds. Unlike the AAPA, the claimed invention GUI is built *before* queries are sent out to the network-connected devices.

With respect to the *first prima* facie requirement needed to support a case of obviousness, there must be some suggestion to combine the prior art references in a manner that makes the claimed invention obvious. If there is a motivation to combine the AAPA and Carcerano

references based upon their common use of conventional device discovery processes, any modifications suggested by the combination would not make the limitations of claims 1 and 15 obvious. In fact, the combination of references can be said to point away from the claimed invention.

"(A)n applicant may rebut a prima facie case of obviousness by showing that the prior art teaches away from the claimed invention in any material respect." *In re Geisler*, 116 F.3d at 1469, 43 USPQ2d at 1365 (quoting *In re Malagari*, 499 F.2d at 1303, 182 USPQ at 553). Here, both the AAPA and Carcerano describe conventional discovery procedures that build a database or GUI only after attempting to query (discover) connected devices. Therefore, the combination of references merely reinforces convention.

The second *prima facie* requirement addresses the same issue from another point of view. Even if an expert were given the two references are a starting point, there is no reasonable expectation that this expert would come up with the claimed invention. Carcerano does not address the network time-out discovery problem. The AAPA mentions the problem, but proposes no solution. Since neither of the references describes a solution to the time-out problem, it is difficult to image how the combination of references describes a solution.

With respect to the third requirement to support a *prima facie* case of obviousness, the combination of references does not teach all the limitations of claims 1 and 15. As noted above in response to the anticipation rejection, claims 1 and 15 recite building a GUI representation of network-connected devices, and only after building the GUI, sending queries to the devices to determine their status.

Both Carcerano and the AAPA only describe building a GUI after all the device query responses are received. Thus, the combination of the AAPA with Carcerano does not explicitly teach all the limitations of claims 1 and 15. Neither do the references suggest any modifications that make these claims obvious. Claims 6 and 8-11, dependent from claim 1, and claims 20, 22-24, and 26, dependent from claim 15, enjoy the same distinctions.

SUMMARY AND CONCLUSION

It is submitted that for the reasons pointed out above, the claims in the present application clearly and patentably distinguish over the cited references. Accordingly, the Examiner should be reversed and ordered to pass the case to issue.

The fee for this Appeal Brief has already been paid. Authorization is given to charge any deficit or credit any excess to Deposit Account No. 502,033.

Respectfully submitted,

Date: 10/1/2006

Gerald Maliszewski
Registration No. 38,054

Customer Number 55,286
P.O. Box 270829
San Diego, CA 92198-2829
Telephone: (858) 451-9950
Facsimile: (858) 451-9869
gerry@ipatentit.net

TABLE OF CONTENTS

CLAIMS APPENDIX

1. (Previously Presented) In a network of devices, a method for a querying device to determine the availability of network-connected devices, the method comprising:

at a querying device, building a graphical user interface (GUI) representation of network-connected devices, prior to sending a query to network-connected devices;

following the building of the GUI, sending a query from the querying device to the network-connected devices;

in response to the queries, updating the GUI representation of the network-connected devices.

2. (Previously Presented) The method of claim 1 further comprising:

at a querying device user interface, issuing a network discovery command; and

wherein building the GUI includes building the GUI in real-time, in response to querying device user interface network discovery command.

3. (Previously Presented) The method of claim 2 wherein building the GUI includes initially representing each of the network-connected devices as unavailable.

4. (Previously Presented) The method of claim 3 wherein sending the query to the network-connected devices includes

spawning a thread from the querying device to query each of the network-connected devices; and

the method further comprising:

receiving a query reply from a first network-connected device; and

wherein updating the GUI representation includes changing the GUI representation of the first network-connected device to available.

5. (Previously Presented) The method of claim 4 further comprising:

failing to receive a query reply from a second network-connected device; and

wherein updating the GUI representation includes maintaining the GUI representation of the second network-connected device as unavailable.

6. (Previously Presented) The method of claim 5 wherein failing to receive a query reply from the second network-connected device includes:

accepting a timeout period for the second network-connected device query; and

if the timeout period expires before a query reply is received, determining that the second network-connected device is unavailable.

7. Canceled

8. (Previously Presented) The method of claim 6 wherein spawning a thread from the querying device to the network-connected devices includes using a function selected from the group including a Sockets connect function, a ping function, and a NSLookup function.

9. (Previously Presented) The method of claim 6 wherein spawning a thread from the querying device to the network-connected devices includes requesting a True/False answer;

wherein receiving a query reply from the first network-connected device includes returning a True answer; and

wherein changing the GUI representation of the first network-connected device to available includes changing the GUI representation to available in response to a True answer.

10. (Previously Presented) The method of claim 9 further comprising:

returning a False answer if the timeout period expires before a query reply is received for the second network-connected device; and

wherein maintaining the GUI representation of the second network-connected device as unavailable includes maintaining the GUI representation as unavailable in response to the False answer.

11. (Previously Presented) The method of claim 10 wherein building the graphical user interface (GUI) representation of network-connected devices includes building a GUI on a computer with a graphical interface; and

wherein spawning a thread from the querying device to the network-connected devices includes requesting the availability of network-connected devices selected from the group including printers, copiers, scanners, faxes, automatic teller machines (ATMs), remote sensors, virtual private network (VPN) devices, satellite devices, and other computers.

12. (Previously Presented) The method of claim 1 further comprising:

accepting a periodic refresh command; and

wherein building the GUI representation of network-connected devices includes refreshing the GUI in response to a refresh command.

13. (Previously Presented) In a network of connected devices, a method of building a graphical user interface (GUI) representing the availability of the network-connected devices independent of system timeouts, the method comprising;

from a querying device, building a graphical user interface (GUI) representation of network-connected devices initially representing network-connected devices as unavailable;

sending a query to a network-connected device; and

modifying the GUI representation of the network-connected device in response to sending the query.

14. (Previously Presented) The method of claim 13 wherein building the GUI includes initially representing the network-connected device as unavailable;

the method further comprising:

receiving a query reply from the network-connected device; and,

wherein modifying the GUI representation includes representing the network-connected device as available in response to the query reply.

15. (Previously Presented) In a network of connected devices, a system for displaying network device availability, the system comprising:

a querying device having a graphical user interface (GUI) representing network-connected devices, the querying device having a network connection port;

at least one device having a network connection port for communications with the querying device; and

wherein the querying device sends a query to5 network-connected devices, after building the GUI, and updates the GUI representation of the network-connected devices in response to sending the queries.

16. (Previously Presented) The system of claim 15 wherein the querying device has a user interface to accept commands; and

wherein the querying device builds the GUI in real-time, in response to commands from the querying device user interface.

17. (Original) The system of claim 16 wherein the GUI initially represents each of the network-connected devices as unavailable.

18. (Previously Presented) The system of claim 17 wherein the querying device spawns a thread to query each of the network-connected devices, and in response to receiving a query reply from a first network-connected device, changes the GUI representation of the first network-connected device to available.

19. (Previously Presented) The system of claim 18 wherein the querying device maintains the GUI representation of a second network-connected device as unavailable, in response to not receiving a query reply from the second network-connected device.

20. (Previously Presented) The system of claim 19 wherein the querying device further includes an operating system and a timer configured with a default timeout value;

wherein the querying device maintains the GUI representation of the second network-connected device as unavailable, in response to not receiving a query reply, as follows:

starting the timer at the beginning of each network-connected device query; and

if the timeout period expires before a query reply is received from the second network-connected device, determining that the second network-connected device is unavailable.

21. Canceled

22. (Original) The system of claim 20 wherein the querying device spawns a thread to query each of the network-connected devices by using function selected from the group including a Sockets connect function, a ping function, and a NSLookup function.

23. (Previously Presented) The system of claim 22 wherein the querying device GUI requests a True/False answer in response to each network-connected device query;

wherein the querying device GUI receives a True answer from the first network-connected device; and

wherein the querying device GUI changes the representation of the first network-connected device to available in response to a True answer.

24. (Previously Presented) The system of claim 23 wherein the querying device generates a False answer in response to a the timeout period expiring before a query reply is received for the second network-connected device; and

wherein the querying device GUI maintains the representation of the second network-connected device as unavailable in response to the False answer.

25. (Original) The system of claim 15 wherein the querying device is a computer and the GUI is represented on a visual display attached to the computer; and

wherein the network-connected devices are selected from the group including printers, copiers, scanners, faxes, automatic teller machines (ATMs), remote sensors, virtual private networks (VPNs), satellite devices, and computers.

26. (Previously Presented) The system of 20 wherein the timer is configured with a refresh rate value; and

wherein the querying device accepts commands for spawning threads to network-connected devices at the refresh rate value; and

wherein the querying device refreshes the GUI, in real-time, in response to the refresh rate value.

EVIDENCE APPENDIX

ATTACHMENT A

RELATED PROCEEDINGS APPENDIX

NONE